Classification of Agricultural Products from X-ray Images

David Casasent and Ashit Talukder

Carnegie Mellon University, Pittsburgh, PA

Thomas F. Schatzki, Pamela M. Keagy and Lan Chau Le

Western Regional Research Center ARS, USDA

August 31, 1999

Abstract

Classification of real-time X-ray images of randomly oriented touching pistachio nuts is discussed. The ultimate objective is the development of a system for automated non-invasive detection of defective product items on a conveyor belt. We discuss the extraction of new statistical features from X-ray images for discrimination between damaged and clean items (pistachio nuts). These features are used as inputs to a new modified k nearest neighbor classifier. This classifier is shown to provide good classification at the desired performance levels.

Key Words: Classification, detection, feature extraction, k nearest neighbor classifier (modified), product inspection, X-ray sensors

INTRODUCTION

The internal product detail that X-ray images provide allows the presence of worm damage and other defects to be determined by non-destructive(non-invasive) methods in various agricultural products, such as appples (1) and other agricultural products (2,3). Current standard inspection techniques cannot determine many defects that X-ray images can, such as worms, insect damage, frass, etc. that are typically not visible through visual inspection. Prior work (3) has confirmed that X-ray images can be useful for identification of infested versus good pistachio nuts.

Prior work on pistachio nut classification (3,4) did not address the problem of segmentation of touching nuts. The image preprocessing developed by us to segment the touching pistachio nuts is discussed in a previous publication (5). In (4) only a hand-picked selected subset of the pistachio nut database was used, and a limited set of histogram features were extracted from each image. In (3), similar features such as the ones used in this paper were used for classification of a smaller version of our database. The current paper uses a much larger database.

Fig. 1a shows a typical image of such products on a conveyor belt that would be input to an automatic vision inspection system. Fig. 1b shows X-ray images of several isolated and touching pistachio nuts in Fig. 1a. Fig. 2 shows typical X-ray images of clean and infested pistachio nuts. As seen, such X-ray images contain useful internal information about the quality of the nut. Infested pistachio nut images (Figs. 2e-h) tend to have large gray scale variations (due to the presence of worm tunnels, infestations, etc.) along with large airgaps (outer dark region between nutmeat and shell), whereas clean nuts typically have smooth nutmeat regions with smaller airgaps (Figs. 2a-d).

This product inspection application is a *very difficult classification problem*; prior work (6, 7) on classification of a subset of these X-ray images by human experts yielded an average P_C correct classification by five of the best human experts of only 86.2% and the median P_C=84% (standard deviation in P_C was 4.3). Therefore, an automated classification method with a P_C performance of $\simeq$ 86% or $\geq$ 86% on the X-ray images of the pistachio nuts is desired.

Ready-for-sale nuts such as those used here, contain 3% of insect or feeding damage, while USDA standards require 1-3% of such damage (6). A 50% reduction in damage, while restricting rejection of good nuts to a commercially

acceptable 1% or less, would be desirable for quality and aflatoxin reduction. Tests show that present automated and manual methods result in rejection of *as much as 20% of the good crop to reduce insect infestations*. Therefore, a better classification method is necessary for this problem.

In this paper, we discuss new rotation and scale invariant histogram features extracted from the the X-ray images that contain better discriminatory information compared to ones used in prior work (4). A new modified k nearest neighbor classifier is used to classify each X-ray image from the extracted features; the modified k nearest neighbor classifier (modified k-NN classifier) is shown to be more robust to the choice of k than the standard k-NN classifier (8). A procedure to correctly classify $\simeq$99% of the clean nuts and locate as much of the infested nuts at the same time is detailed. We also discuss a scheme to select the number of histogram features to use from training data P_C performance. This scheme is compared to to that obtained using the orthonormal discriminant vector (ODV) procedure (9,10) (also the Fisher linear discriminant (11) which is a subset of the ODV) that is computed from *all* histogram features.

The database used and the new histogram features extracted from the the X-ray images are detailed first. The new modified k nearest neighbor classifier to classify each X-ray image from the extracted features is discussed next. We discuss the method we use to order the histogram features by their importance or usefulness for discrimination and the selection of the *number* of the most important histogram features to use. The P_C performance using the ODV features computed from the input histogram features is also discussed.

DATABASE USED

The database we consider consisted of 25 trays of scanned 2-D X-ray film images of large pistachio nuts (18-20 nuts per ounce). Each image contained about 100 nuts at random orientations. The nuts used were obtained from a processor after sorting and sizing processing and hand-inspection to remove twigs and other non-nut material. These are typical images of such products on a conveyor belt that would be input to an automatic vision inspection system.

Each tray was X-rayed (90 sec. at 25 keV [with an 0.25 mm Be window] with a Faxitron series X-ray system

4380N, Faxitron Corp., Buffalo Grove, IL; Industrex B film, Eastman Kodak, Rochester, NY). Twelve bit digital images of these X-ray films were obtained at a resolution of $(0.173 \text{ mm})^2$/pixel using a Lumiscan 150 film scanner (Lumisys, Sunnyvale, Ca). These images were reduced in pixel count by a factor of nine by pixel averaging to produce 12-bit images with a resolution of $(0.5 \text{ mm})^2$/pixel. These are refered to as *X-ray film images*. Further details of the X-ray imaging system and the digitization technique used is discussed in (5).

All of the nuts were dissected and classified into good and insect damaged (bad) by visual inspection after dissection (6). Nuts in other categories (immature kernel, large kernel spots, etc) were not included. A total of 1884 nuts were used for classification; these were divided into a training set of 942 nuts (600 good and 342 bad) and a test set of 942 nuts (600 good and 342 bad); *only the training set was used to select the classification parameters (such as selecting the number of features to use, and the parameters to normalize each feature before classification).* Figure 2 shows a few of the X-ray images of clean and infested pistachio nuts in the database. To obtain classification accuracy using only the training set data, we used a leave-one out testing classification scheme on the training data. Leave-one-out classification of a training set of size S involves picking one of the training samples in the training set, and using the rest of the $S-1$ training data samples as prototypes to classify the selected sample. This is repeated S times, each time using a different sample; the overall P_C performance on the training set is then noted.

DESIRED PERFORMANCE

As discussed earlier, for this pistachio nut classification problem, obtaining the best overall classification accuracy is not necessary. It is often necessary to obtain better P_C for one class as compared to another class; this is the case in our pistachio nut classification problem. It is preferable that $\geq$50% of the infested pistachio nuts in the database be correctly classified, while correctly classifying $\simeq$99% of the clean nuts in the database.

Receiving operating characteristic (ROC) curves (12) are commonly used for applications where true objects need to be classified versus non-objects (clutter). ROC curves involve plotting the probability of detection (P_D) or the probability of correct classification of an object (P_C) versus the probability of false alarm (P_{FA}) by varying prior probabilities for each class (object versus clutter). In our current application, clean pistachio nuts are be viewed as

objects, and infested pistachio nuts are treated as clutter. The ROC curves we use show the variation in the ratio of *correctly classified clean nuts* versus the *ratio of incorrectly classified infested nuts*.

ROTATION AND SCALE INVARIANT HISTOGRAM FEATURES

The features extracted from each input X-ray image should contain useful information that allows discrimination between clean and infested pistachio nuts. These features should be *scale-invariant* since the size of a pistachio nut can vary from as few as 380 pixels to as many as 900 pixels. The input pistachio nuts can lie at *any orientation*. Therefore, *rotation-invariant* features are needed. Histogram statistic features of an input gray-scale image are rotation-invariant since they do not contain spatial information. Histogram statistic features are also scale-invariant since they are obtained by dividing the gray-scale distribution on only the pistachio nut pixels by the total area of the nut.

The features we extracted from each pistachio nut X-ray image were the mean, variance, skewness, and kurtosis (histogram features) of *four* different images of each nut (raw, edge, curvature (13) of raw and curvature of edge images) (14). Variations of these features have been used in prior X-ray pistachio nut classification work (7). In prior work (4) on a handpicked subset of the database used in this thesis, histogram features were extracted from only the input raw X-ray image and the edge enhanced X-ray image. The edge images were obtained by calculating the sum of the absolute values of the differences between a given pixel value and its eight neighbors (4). For each of the resultant four sets of images of the segmented nuts, four sets of histogram features were calculated. This gave a total of 16 possible features. The histogram for each image of each nut was divided by the total number of pixels in each nut image (this provides scale-invariant features). The four histogram features were then calculated separately for the four sets of images. The mean measures the average gray value on the pistachio nut, and variance is a measure of "spread" of gray values within the nut. Skewness is a measure of the symmetry of the distribution; kurtosis is a measure of how sharply peaked the distribution is.

The raw images for infested pistachio nuts tend to be darker with more gray-scale variations, whereas clean nuts tend to have smoother gray values. Hence, we expect the variance of the raw images to be different for infested

pistachio nuts. Edge (high-frequency) information is also expected to be useful since infested pistachios tend to have rougher texture compared to clean nuts. The edge images for a clean (Figure 3a) and infested (Figure 3e) pistachio nut are shown in Figs. 3b and 3f respectively. The curvature images provide information about the rate of change of gray-values (combination of first and second order differentials in gray-scale) over the entire gray-scale pistachio image. This information is very useful, especially in regions near the airgaps. Infested pistachio nuts tend to have larger and darker airgaps and sharp transition regions (high curvature) between the airgap and nutmeat, whereas clean pistachio nuts tend not to have such high gray-level transition (curvature) regions. Therefore, we expect the curvature image to contain important discriminatory information about each individual pistachio nut. A curvature image for a good (infested) nut is shown in Figure 3c (Figure 3g). The curvature image of the edge-enhanced image for a clean nut and an infested one are shown in Figs. 3d and 3h respectively; no clear differences are immediately apparent.

For the curvature and edge-enhanced images, we erode the output images with a 3×3 structuring element prior to using them; this removes the outer boundary between the shell and the background which tend to have very high curvature values due to the gray-scale change in such regions. For only the curvature images, the output images were clipped at $\pm T_C$ (all values $\geq T_C$ are set to T_C and all values $\leq -T_C$ were set to -T_C; T_C=1.5 was used) and each clipped curvature image was separately normalized to 0-255. This separate normalization reduces the usefulness of the mean of a nut in curvature data but enhances the use of variance, skewness and kurtosis, since good nuts should not have many concave regions (worm tunnel etc.) while bad nuts should have both concave and convex regions.

CLASSIFIER

Modified k-Nearest Neighbor Classifier

A *new modified nearest neighbor algorithm* was used to classify sample data using features extracted from each sample. The standard nearest neighbor classifier (8) classifies test data by comparing the Eucledian distance of each test sample to the nearest prototype (training) sample. The test sample is assigned to the class of the closest prototype sample. A variation of the nearest neighbor classifier is the k-nearest neighbor (k-NN) classifier. In this

method, the k nearest prototype samples from a test sample are computed. The test sample is then assigned to that class with the majority (winning class) among the k nearest prototype samples. The k nearest-neighbor classifier has been proven to provide classification with a maximum classification error that is twice that of the Bayes classifier (8).

However, the k-NN has problems when the amount of overlap between classes is high, and the number of prototypes (training samples) per class is low (15). When the number of prototypes is low, the classification results depend highly on the choice for k (15, 16). The distance to the nearest prototype samples is a useful measure when the number of prototypes in each class is low, and such information could provide a more robust measure for classification. Such modifications have been suggested for the nearest neighbor rule (16–18). A weighted distance k-NN technique (16) weights each of the k-nearest neighbors based on the distance to the test sample; the weight assigned for each of the k nearest samples is inversely proportional to the distance to the test sample. For each class among the k-nearest samples, the sum of the weights for that class is computed. The class with the largest sum of weights is assigned as the winning class. A nearest unlike neighbor scheme has also been suggested (17) in which the relative distance between the winning nearest neighbor class and the closest losing class (nearest neighbor among the other classes) is used as a confidence measure for classification. This approach is used to reject outliers in the test data; however, the nearest unlike neighbor scheme uses only k=1.

The standard k-NN classifier also has problems for test samples that lie close to the boundary between different classes. In such cases, the choice of k in the k-NN classifier is a critical parameter. An example is shown in Figure 4a, where the test sample (+) belonging to class 1 (o) is located in a region of overlap between class 1 (o) and class 2 (×). If k is small (k<3) for this example (Figure 4a), the test sample (+) will be classified wrongly as a class 2 (×) sample. An example where a large value of "k" for the k-NN will result in the test sample from class 1 (+) being wrongly classified as belonging to class 2 (×) is shown in Figure 4b. In this case, the 3 nearest samples to the test sample belong to the correct class 1 (o), but when k=7,9 or 11, the test sample in Figure 4b will be wrongly classified as a class 2 (×) sample. As shown in these examples, the *k-NN can be quite sensitive to the choice of k*, especially for samples that lie close to the decision boundary between two classes. The reason for this is that the k-NN does not use the *distance* of each test sample from the k-nearest prototypes.

We use a modified k-NN using the closest average distance per class for the k-nearest neighbors for each class.

This approach appears to be robust to the presence of outliers in the prototype data. Our method is similar to the weighted distance technique (16), but is simpler since we use a linear function of the distance to the closest prototypes. The modified k nearest-neighbor classifier we use is now detailed. For each test sample, we compute the *average distance* of each test samples to the closest k samples *in each class.* Thus, we calculate: the average distance of the test sample to the k closest prototype samples in class 1 ($d^1_{kavg} = (1/k)\sum_{i=0}^{k}(\underline{x}_t - \underline{x}^1_i)^2$), the average distance to the closest prototypes in class 2 ($d^2_{kavg} = (1/k)\sum_{i=0}^{k}(\underline{x}_t - \underline{x}^2_i)^2$), etc. for all L classes. The test sample is assigned to the class with the closest average distance, d^l_{kavg}, i.e., the test sample is assigned to class c if $d^c_{kavg} < d^l_{kavg}$ $\forall l = 1, 2...L, l \neq c$. When k=1, each test sample is classified to the class corresponding to the nearest neighbor and our modified k-NN is the same as the nearest-neighbor classifier. For training data, we use the leave-one-out procedure discussed in Section to obtain the P_C performance of the modified k-NN on the training set.

There are two main advantages to our new modified k-nearest neighbor classifier compared to a k-NN classifier. The k-NN is sensitive to the choice of k for test samples that lie close to class boundaries when the number of prototypes is small (15) (Figure 4). If our modified k-NN is used for the examples in Figures 4a and 4b, the test samples in Figures 4a and 4b will be correctly classified as a class 1 sample (o) for a reasonable choice of k (k=4 to 7 for Figure 4a and k=1 to 7 for Figure 4b). This occurs because the average distance of the test sample to the class 1 prototypes will be smaller than the average distance of the test sample to the class 2 prototypes in both Figures 4a and Figures 4b. Therefore, *we expect our modified k-NN to be robust for test samples that lie close to the decision boundary* between two classes, since it rejects outliers in the prototype data.

We noted earlier (Section) that *it is not always preferable to obtain the best overall classification accuracy (P_C) for both classes.* The ROC curve of P_C for clean nuts versus the ratio of incorrectly classified infested nuts with the standard k-NN classifier is obtained by changing the desired "majority" for each class (18). Since the majority measure in the k-NN can only be changed in integer increments, only coarse ROC measurements result; *another minor advantage of our modified k-NN is that it provides finer ROC curves.* To obtain an ROC curve using our modified k-NN method, we assign a test sample to a class using the rule: the test sample is assigned to class c if $(d^c_{kavg} - d^l_{kavg}) < T_c$ $\forall l = 1, 2...L, l \neq c$ where we vary the T_c threshold for class c. If $T_c = 0$, each sample is assigned

to the class with the closest average distance d^c_{kavg}. If $T_c = +\infty$, all test samples are assigned to class c; if $T_c = -\infty$, then none of the test samples are assigned to class c. Therefore, *the thresholds T_c are analogous to the confidence measure assigned to each class.*

Parameter (k) Selection in Modified k-NN Classifier

Not much prior work has been done on the optimal choice of k in the k-NN classifier. It has been noted (15) that the choice of the optimal k in the k-NN classifier is a function of three parameters: feature dimensionality, number of prototypes, and the ratio between the number of samples in both classes. The choice of the optimal k increases with increasing sample size *and* increasing feature-dimension size; for a given training sample size and feature dimensionality, the P_C performance typically increases with increasing k up to some k value and then decreases (16). This property has been theoretically proven for noise-free Boolean valued features called Boolean threshold functions (15). This is often referred to as the *peaking performance* of the k-NN by empirical analysis (15,16). We expect the same trend to hold for our modified k-NN classifier. We determine the optimal choice for k in the modified k-NN classifier by evaluating its performance with changing k on training data (using a leave-one-out test on training data as discussed in Section). We varied k from k=1 to k=23 in the modified k-NN, using all 16 input histogram features (Section). The P_C performance of the modified k-NN classifier is seen (Figure 5) to *increase steadily* with increasing k until k=17, then the P_C performance drops with k >17. We also tested the P_C performance of the standard k-NN classifier with varying k. In keeping with our statement earlier (Section , page 8), we observe (Figure 5) that P_C performance of the standard k-NN classifier is "more erratic" with changing k than that of the modified k-NN classifier. We feel that occurs because the standard k-NN classifier does not perform robust classification of the test samples that lie close to the class boundaries; its performance is therefore sensitive to the choice of k. Therefore, the modified k-NN is preferable in this case.

The P_C performance of the modified k-NN on the training data was found to peak at k=17 (Figure 5). Therefore, we used a value of k=17 in the modified k-NN classifier to classify the pistachio nuts using different feature spaces in Section . Note however that the choice of k in the modified k-NN is not critical and gives comparable P_C results for other values of k in the modified k-NN.

CLASSIFICATION RESULTS

The 16 rotation and scale-invariant histogram features (Section) extracted from each individual pistachio nut are not all equally useful for discriminating between clean and infested pistachio nuts. Hence, it may be necessary to order these features by their importance for discrimination and to select a subset of these features as inputs to the classifier (classifiers themselves *cannot easily find both the features and the best combination of features to use*). The selection of the optimal subset of features to use for a specific application is best done by evaluating P_C peformance of *all possible subsets of input features*. It is well known that the problem of selecting the best subset of input features to use for classification is an NP-complete problem (19). We use a sub-optimal solution where we use SAS (20) to order the input histogram features. The number of these ordered features to use is determined from the P_C performance on the training set (using a leave-one-out procedure). The forward-selection technique (20, page 911) in the STEPDISC procedure in SAS (20) was used to order the input histogram features in their order of importance for discrimination. The default parameters (20) of the stepwise discriminant analysis procedure were used.

The 7 histogram features selected by SAS were (in order of importance): Variance of curvature of raw images, Kurtosis of raw images, Variance of curvature of edge images, Variance of raw images, Variance of edge images, Skewness of curvature of raw images, and the Mean of curvature of raw images.

We tested the classification performance using two sets of feature spaces. In ***both* cases, the modified k-NN classifier was used**. We first classified each pistachio nut in the test set using the original 16 histogram features (4 statistical features each from the raw, edge, curvature, and curvature of edge image). Next, we classified the images using ODV (and Fisher linear discriminant) features (computed from the original histogram features). We show that comparable classification is obtained using ODV features computed from *all* input histogram features as compared with using the best subset of histogram features (Section).

Classification Using Original Histogram Features

The 16 original histogram features were analyzed using SAS; forward-selection was used to order the histogram features. Each feature was normalized to a zero to one range (using only the training set data). Test set data was normalized using the training set parameters. The first seven histogram features selected by SAS were (in order of importance): variance of curvature of raw images, kurtosis of raw images, variance of curvature of edge images, variance of raw images, variance of edge images, skewness of curvature of raw images, and the mean of curvature of raw images. Table 2 shows P_C (percentage of correct classification) obtained with different numbers of features (ordered by SAS) using the modified k-NN classifier with a high value of k=17. As the number of histogram features used was increased, the percentage of nuts correctly classified P_C generally increased and rapidly became fairly constant (Table 2). From training set data, we determined to use seven histogram features (see bold entries in Table 2). P_C performance improved and then dropped as the number of features added was increased. P_C results for the test set follow training set data quite well in Table 2, thus, generalization is good. Therefore, there are difference in the P_C performance when the number of input features vary from 2 to 16.

Preferable P_C Measure Classifier

Tables 1.1 and 1.2 show the confusion matrices for the training and test sets using the modified k-NN with k=17, and the seven best original histogram features. Both training and test set scores are similar; this shows good generalization. As noted in Section , *the standard P_C performance measure is not the desired one, nor is a classifier with the best P_C* (as seen in Table 1.2, this will reject around 5.5% of the good crop while rejecting 79.2% of the bad crop). Therefore, while overall P_C performance is optimal, a large number of clean pistachios are rejected. To achieve preferable performance (Section), we consider detecting only infested nuts with a high degree of confidence.

The algorithm we use is now described. We only classify a nut as infested (bad) if the average distance of each test sample to the k-nearest *infested* pistachio nut prototypes is less than the average distance to the k-nearest *clean* (good) nut prototypes by T. Using this technique, only infested nuts with high likelihood of correct classification (large T) are removed and very few clean nuts are expected to have a large T and be misclassified as bad and be rejected. Note that varying T is similar to varying the ratio of the prior probabilities for each pistachio nut class in

a statistical classifier.

Table 3 shows new preferable P_C results as T is varied. As T is decreased, more bad nuts are detected (P_C (bad) increases) but more good nuts are rejected (P_C (good) decreases). Thus, a larger T is preferable. T=0.0325 results in classification (rejection) of $P_C \simeq 52\%$ of the bad nuts in the training data (*this reduces the bad nuts to 1.5% of the crop) while rejecting only 1% of the good nuts (P_C=99% for good nuts*). We use the same threshold T value for the test set. The performance of the test set for this preferred operating point follows the training set; $P_C \simeq 49\%$ of the bad nuts in the test set are rejected (correctly classified), while rejecting only 0.7% of the good nuts. It is also possible to select other operating points in Table 3, if it is desired to locate more bad pistachio nuts, at the cost of lower P_C performance for good nuts.

Classification using ODV Features

For purposes of comparison, we also extracted ODV features from the five and seven best original histogram features, and from all 16 original histogram features. The classification results using ODV features are shown in Table 4. Note that when only one ODV feature is used, the ODV is exactly the same as the Fisher linear discriminant. The number of ODV features to use was determined using the training set (bold text in Table 4). The best P_C results using ODV features were *comparable* to those obtained using the seven best histogram features.

SUMMARY

We have obtained excellent classification results on segmented agricultural products using new rotation and scale invariant features, and a new modified k-NN classifier. Use of our histogram features was shown to provide better classification than the average P_C performance by the five best human experts. An improvement of $\simeq 2.4\%$ was obtained over the average performance of the human subjects. *This improvement is commendable since many of the misclassified infested (clean) nuts do not look bad (good) (from X-ray images); e.g. nuts with splits can be misinterpreted as having worm tunnels, airgaps could be classified as infested regions, etc.* The truthed data provided was obtained by dissecting each nut, and visually classifying each dissected nut. Much of this information visible

after dissection does not show up in the X-ray imagery. Hence, this is a formidable pattern recognition problem, and even 2-3% improvements in classification at the 88% classification level is notable. Despite these problems, the classification achieved here, 50% of bad nuts recognized with 1% of good nuts falsely rejected, appears to be very useful.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the work of Lan-Chau Lee for categorization of the pistachio nuts. The classifier work is supported by a recent 1998 U.S. Department of Agriculture grant. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the U.S. Department of Agriculture.

References

1 T. Schatzki, R. Haff, R. Young, I. Can, L. Lee, and N. Toyofuku, Defect detection in apples by means of x-ray imaging, *Proceedings of the Society of Photo-Optical Instrumentation Engineers* **2907: paper 18** (1996).

2 P. M. Keagy and T. Schatzki, Machine recognition of weevil damage in wheat radiographs, *Cereal Chemistry* **70**, 696–700 (1993).

3 P. M. Keagy, B. Parvin, and T. Schatzki, Machine recognition of navel worm damage in X-Ray images of pastachio nuts, *Lebensm-Wiss U Technol* **29**, 140–145 (1996).

4 D. Casasent, M. A. Sipe, T. F. Schatzki, P. M. Keagy, and L. L. Lee, Neural net classification of X-ray pistachio nut data, *Lebensm.-Wiss. u.-Technol.* **31**, 122–128 (1998).

5 D. Casasent, A. Talukder, P. Keagy, and T. Schatzki, Detection and Segmentation of Items in X-Ray Imagery, *Tentatively Accepted in Trans. ASAE* (1998).

6 P. M. Keagy, B. Parvin, T. Schatzki, L. Le, D. Casasent, and D. Weber, Expanded image database of pistachio X-Ray images and classification by conventional methods, *Proc. SPIE*, **2907**, 196–204, (1996).

7 P. M. Keagy, T. F. Schatzki, L. L. Lee, D. Casasent, and D. Weber, Classification of X-ray pistachio nut images by conventional parametric methods, *Submitted to Lebensm.-Wiss. u.-Technol.* (1999).

8 T. M. Cover and P. E. Hart, Nearest neighbor pattern classification, *IEEE Trans. on Information Theory* **IT–13**, 21–27 (1967).

9 D. H. Foley and J. W. Sammon, An optimal set of discriminant vectors, *IEEE Trans. Comput.* **C-24**, 281–289 (1975).

10 Y. Hamamoto, T. Kanaoka, and S. Tomita, On a theoretical comparison between the orthonormal discriminant vector method and discriminant analysis, *Pattern Recognition* **26**, 1863–1867 (1993).

11 R. A. Fisher, *Contributions to Mathematical Statistics*,New York: John Wiley, (1950).

12 H. L. Van Trees, *Detection,estimation, and modulation theory*,New York: Wiley, (1968).

13 I. Faux and M. Pratt, *Computational Geometry for Design and Manufacture*,Horwood, NY: Halstead Press, (1979).

14 D. Casasent, A. Talukder, and H.-W. Lee, X-Ray Agricultural Product Inspection: Segmentation and Classification, *Proc. SPIE*, **3205**, 46–55, (1997).

15 S. Okamoto and K. Satoh, An average-case analysis of k-nearest neighbor classifier, *Intl. Conf. on Case-Based Reasoning and Development*, 253–264, (1996).

16 S. A. Dudani, The distance weighted k-NN rule, in B. V. Dasarathy, ed., *Nearest Neighbor Norms: NN pattern classification Techniques*Los Alamitos, CA: IEEE Computer Society Press, (1991), 92–94.

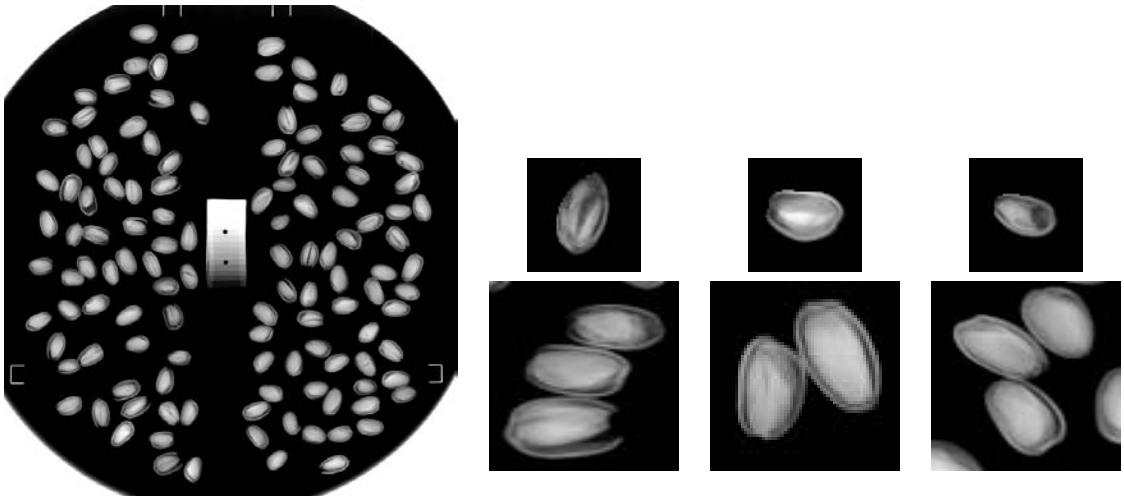
17 B. V. Dasarathy, Nearest unlike neighbor (NUN): An aid to decision confidence estimation, *Optical Engineering* **34**, 2785–2792 (1995).

18 I. Tomek, A generalization of the k-NN rule, in B. V. Dasarathy, ed., *Nearest Neighbor Norms: NN pattern classification Techniques*Los Alamitos, CA: IEEE Computer Society Press, (1991), 86–91.

19 K. S. V. Horn and T. R. Martinez, The minimum feature set problem, *Neural Networks* **7**, 491–494 (1994).

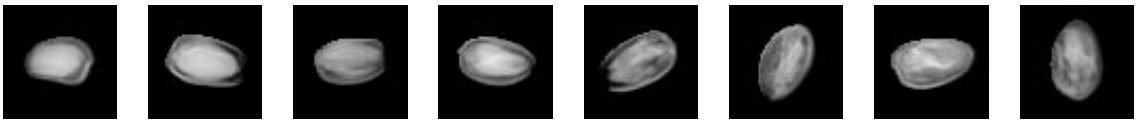
20 *SAS/STAT Users Guide: Release 6.03*,Cary, NC: The SAS Institute, (1988).

List of Figures



(a) (b)

Figure 1: Typical scanned X-ray image of a tray of pistachio nuts (a) and individual nuts (b) showing internal detail



(a) (b) (c) (d) (e) (g) (g) (h)

Figure 2: X-ray images of clean (a-d) and infested (e-h) pistachio nuts.

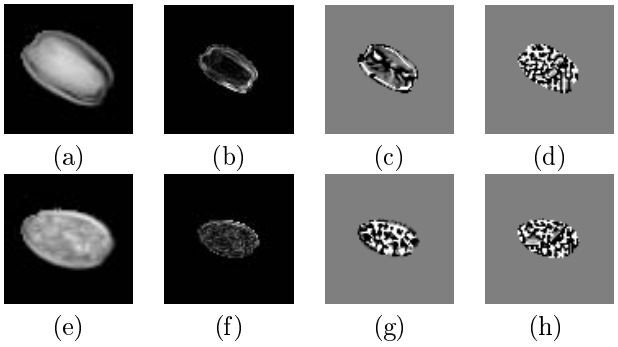
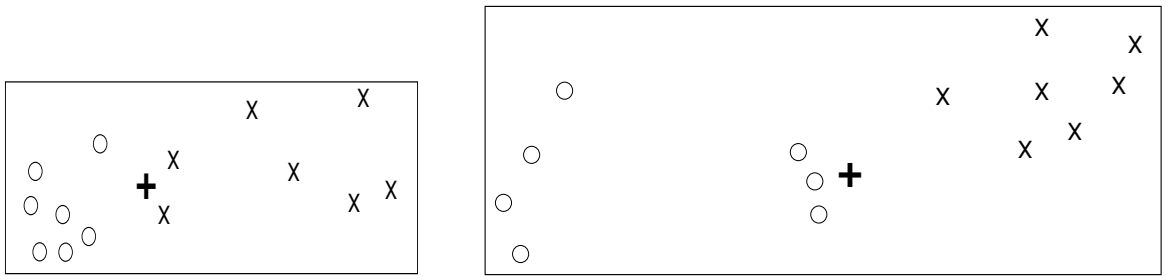


Figure 3: (a) Raw image of clean pistachio nut, (b) corresponding edge image, (c) corresponding curvature image, and (d) corresponding curvature of edge image; (e) Raw image of infested pistachio nut, (f) corresponding edge image, (g) corresponding curvature image, and (h) corresponding curvature of edge image.



(a) (b)

Figure 4: Examples exhibiting sensitivity of the k-NN to the choice of k: (a) k-NN fails with small k, (b) k-NN fails with large k. In both cases, the test sample '+' belongs to class '*o*'.

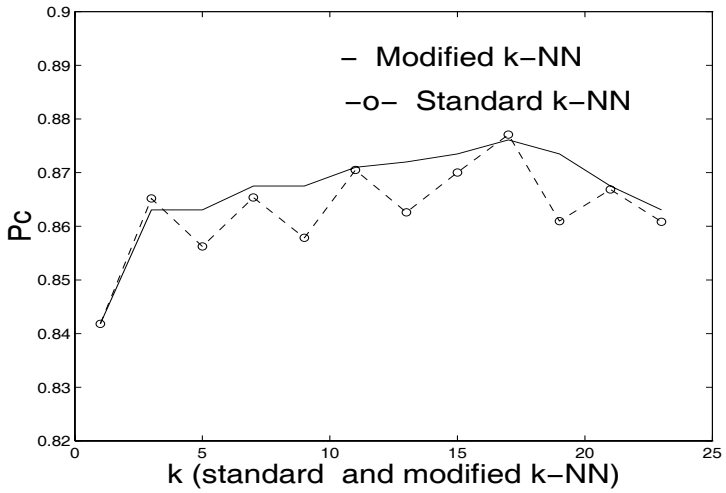


Figure 5: P_C on training data with changing k in standard k-NN and modified k-NN for all input histogram features.

	Good	Bad
Good	93.7%	6.3%
Bad	12.5%	77.5%

Table 1.1: Confusion matrix using modified k-NN (Training set, 7 features).

	Good	Bad
Good	94.5%	5.5%
Bad	10.8%	79.2%

Table 1.2: Confusion matrix using modified k-NN (Test set, 7 features).

Classifier	No. of Features	1	2	3	4	5	6	7	16
Modified	P_C (Train)%	82.3	87.2	87.2	87.7	87.8	87.9	**88.0**	87.6
k-NN (k=17)	P_C (Test)%	80.1	86.9	86.6	87.7	87.7	88.4	**88.7**	87.4

Table 2: P_C for different numbers of histogram features (modified k-NN). The P_C confidence interval bounds are $\pm 2.15\%$ at 87%, $\pm 2.1\%$ at 88% and $\pm 2\%$ at 89%.

T	0.0325	0.0215	0.0110
P_C(Good) Train	**99**%	98%	97%
P_C(Bad) Train	**51.8**%	58.8%	70.2%
P_C(Good) Test	**99.3**%	99%	97.5%
P_C(Bad) Test	**48.8**%	57.3%	68%

Table 3: Classification accuracy with varying T. The confidence interval bounds for the P_C for infested nuts only are $\pm 5.3\%$ at 48.8%, $\pm 5.2\%$ at 57.3% and $\pm 4.9\%$ at 68%.

Original Histogram Features used	ODV features used	1	2	3	4	5
		P_C (%)				
5 Best	Train	88.1	**88.3**	87.8	-	-
Original Histogram Ftrs	Test	87.3	**87.2**	86.8	-	-
7 Best	Train	88.2	**88.4**	87.6	87.5	87.9
Original Histogram Ftrs	Test	87.7	**88.2**	87.4	87.6	87.2
16 Best	Train	87.4	**88.5**	88.2	87.4	87.8
Original Histogram Ftrs	Test	88.6	**88.6**	88.8	88.7	88.6

Table 4: P_C for ODV features computed from different subsets of original histogram features (modified k-NN, k=17). The P_C confidence interval bounds are ±2.15% at 87%, ±2.1% at 88% and ±2% at 89%